Cluster Establishment in Vehicular Networks: Connectivity Enhancement

PHAM Thi Ngoc Mai
Supervisor NGUYEN Minh Huong

University of Science and Technology of Hanoi
ICT Department

2020-08-05

Acknowledgement

This thesis project could not have been completed without the intellectual and emotional support from many important people in my life. I am extremely grateful for having Dr. Nguyen Minh Huong as my supervisor. Her patience, wisdom, understanding and encouragement inspired and guided me through some of the most challenging periods and pushed me farther than I though I could do. To all my teachers in ICT Department, thank you for the knowledge and experience you conveyed to me during 3 years in USTH so that I can have solid foundation to carry out this project. Thank you - especially Lien Huong, Khanh Huyen, Do Ngoc, Bach Pham, Chu Hiep, Do Hoang for being with me during my university years as well as giving comments for my thesis. To all my friends, labmates and family for encouraging and disciplining me to focus on my work.

List of Acronyms

USTH University of Science and Technology of Hanoi

LID Lowest Identification

HCC Highest Connectivity Clustering

DSRC Dedicated Short Range Communication

WAVE Wireless Access in Vehicular Environment

CH Cluster Head

CM Cluster Member

VANET Vehicular Ad-hoc Network

MANET Mobile Ad-hoc Network

QoS Quality of Service

RSU Road-side Unit

CSMA Carrier-sense Medium Access

PDR Packet Delivery Ratio

List of Tables

List of Figures

Contents

I. Introduction

1.1 Context and Motivation

The application of information technology appears in every aspect of life. Nowadays, vehicles are not only equipped with powerful engine but also safety-awareness applications and communication capabilities thanks to the advances in wireless network. However, most of the safety systems are based on the information collected from sensors without the communication with vehicles surroundings. This makes a number of vehicles difficult to respond to urgent problem nearby. The demand of inter-vehicle communication led to the emergence of vehicular ad-hoc network (VANET).

VANET was introduced by Intelligent Transportation Systems in order to overcome issues from daily transportation (traffic accidents, road congestion, fuel consumption,…) and make the journey safer and entertaining. It is a subclass of mobile ad-hoc network (MANET) where vehicles are considered as mobile nodes. It provides two types of communication:

- V2V (Vehicle-to-Vehicle): allows vehicles communicate with each other.
- V2I (Vehicle-to-Infrastructure): allows vehicles to communicate with external system such as Road-Side Units (RSUs).

There are mainly two type of applications used in VANETs, safety and non-safety application. Among them, safety applications are responsible for sending safety message, for example, incident notifications, road hazard notifications, emergency vehicle warnings, so that proper actions can be taken to prevent accidents and save people from hazardous situations.

To support those applications, Wireless Access in Vehicular Environment (WAVE) - a vehicular communication system was introduced in IEEE 802.11p. This supports data exchange between high-speed vehicles and between the vehicles and the road-side infrastructures in the licensed ITS band of 5.9 GHz (5.85~5.925). WAVE is the basis for Dedicated Short Range Communication (DSRC) - a communication technology which divides the bandwidth into 7 channels: one control channel (CCH) and 6 service channels (SCHs) particularly designed for applications such as safety service, toll collection and commerce transactions via cars.

In VANETs, messages that contain critical events information (e.g. accident reports) have to be exchanged quickly, reliably and ensured to access to sufficient necessary vehicles. However, with the help of WAVE/DSRC technology, it is still a challenge to disseminate critical message timely and reliably to a targeted area under a dynamic vehicular environment[1]. Several researches have been conducted with the aim of improving data transmission capacity in VANETs concerning control schemes for media access or network topology management. One of those schemes establishes a clustering structure within the network.

Clustering establishment is a way to manage nodes in network hierarchically. The main idea is that nodes with some common characteristics are partitioned in to groups (clusters), each has a cluster head (CH) node which manages the communication with other CHs, coordinates the operation of cluster members (CMs). Organizing VANETs by a clustering protocol help reducing overhead as well as communication failure in vehicular network, supporting QoS and lay the foundation for routing strategies.

In this work, we introduce a clustering protocol which focus on enhancing the connectivity of nodes in vehicular network so that a notification of accidents or any emergency reports has higher coverage over network. This protocol is built on the idea of my supervisor about process-division clustering establishment and beacons message dissemination. I am the one who finalized the protocol and implemented it in simulation tool NS-3, evaluate its performance through basic scenarios for safety applications.

1.2 Objectives

To construct this algorithm and verify its effectiveness in connectivity enhancement, our goals include:

- Give a complete design of the protocol with a thorough description of each process and each type of participant in the network (vehicle or road-side unit)
- Implement the protocol by using network simulator NS-3
- Design scenarios for safety applications including safety message dissemination inside cluster and across clusters
- Run simulation for designed scenarios and analyse the performance on the aspect of connectivity improvement

1.3 Thesis structures

The rest of the thesis is organized as follows: Section 2 provides an overview of the common way to disseminating message within vehicular network (both clustering-based and non-clustering based) that can be found in the literature. Section 3 gives a detailed explanation about our protocol. Section 4 describes the design for implementations and scenarios and tools we used for coding and simulations. The next section shows results and evaluate the performance of the model we proposed. The last section concludes our work.

II. State of the art

In this section, we provide some approaches of data propagation schemes in networks. One of the characteristics of ad-hoc network is decentralization, which means it does not rely on any pre-existing infrastructure such as routers or access points. Nodes participate in routing by only forwarding messages.

Under the situation that there is no protocol supporting for the forwarding process, nodes with re-broadcast message if it is not the destination of the received packet. This blind broadcasting method has a major drawback. It is easily causing the overload and collisions within network since nodes keeps re-broadcast messages endlessly. In order to support packet routing in VANETs, various clustering-based algorithm have been proposed.

Some early well-known clustering algorithms are LID (Lowest ID) and HCC (Highest Connectivity Clustering) [2]. The ideas of two algorithms are simple and easy to understand from their names. LID chooses node with lowest ID to become CH, meanwhile, HCC elects the nodes with highest connectivity (highest number of neighboring nodes). Both methods are lack of mobility-awareness in VANETs. As a result, formed clusters are reconfigured frequently, causing overload and connections betweens CMs are not ensured in long term.

Clustering establishment should be fast but also have to ensure the stability, connectivity and delay so that clusters can operate properly and efficiently after being created. Newer approaches follow a competitive election mechanism, where the mobility of nodes is taken into account.

In [3], the authors proposed a cluster based location routing. Nodes use HELLO messages, which are scheduled to send periodically to distribute their states. Based on information collected from HELLO messages, node can decide wether it will join a cluster and become CM or announce itself as a CH. Nodes store information from HELLO messages into a table, so that they know their neighbors and forward a data packet directly to destination. This approach mainly focus on improving routing efficiency.

In [4] a novel approach introduced with the aim of maintaining cluster stability by introducing new terminology about r-stable neighbors and 2r-stable neighbors. Nodes still use periodic message to exchange their states but they divide their neighbor list into 2 types as mentioned before. Nodes which are 2r-stable neighbors can join the CH election process, but after CH is elected, only nodes in r-stable list can join the cluster formation process. Velocity deviation is the criteria for dividing neighbors into two types.

A cluster-based traffic information generalization in VANETs, as described in [5] proposed a method for CH election by voting. Each node calculate its Befit Factor (BF) and broadcast its BF value and store BF values from neighboring nodes. After that, each node will vote for the node in its neighbor list (or it can vote for itself) with local highest BF value and broadcast its vote. This algorithm is proved to enhance the cluster head duration and reduce the number of clusters created in network.

Clustering establishment is a way to manage nodes in network hierarchically. The main idea is that nodes with some common characteristics are partitioned in to groups (clusters), each has a cluster head (CH) node which manages the communication with other CHs, coordinates the operation of cluster members (CMs). Organizing VANETs by a clustering protocol help reducing overhead as well as communication failure in vehicular network, support QoS and lay the foundation for routing strategies.

In this work, we introduce a clustering protocol which focuses on enhancing the connectivity of nodes in vehicular network so that a notification of accidents or any emergency reports has higher coverage over the network. This protocol is built on the idea of my supervisor about process-division clustering establishment and periodic message dissemination that we call beacons. I am the one who finalized the algorithm for each process and implemented the algorithm in simulation tool NS-3, evaluate the algorithm through basic scenarios for safety applications.

III. Methodologies

3.1 System overview and assumptions

In our system, the behaviors of a node depend on its NodeState or state and the current process. State can be one of three values:

- STANDALONE: when the cycle time is reset, nodes will become STANDALONE which mean they do not belong to any cluster.
- CH: the cluster head, elected from CLUSTER_FORMATION process.
- CM: the cluster member.

In short, most of the proposed clustering protocols aim to increasing cluster's lifetime and reducing changes in network topology. They follow a pattern of two phases: Clustering Establishment and Clustering Maintenance to update network topology. However, we do not follow establishment-maintenance pattern. Instead, nodes are supposed to operate in cycles. Each cycle include 3 process: BEACON_EXCHANGE, CLUSTER_FORMATION and DATA_EXCHANGE.

We limit the operation time for each process, when the time is up, every node have to switch to the next process. At the end of each cycle, nodes abandon all their information and refresh its storage and states to prepare for a new cycle, when the clustering is started over again. Figure 1 gives a visualization of our time cycle.

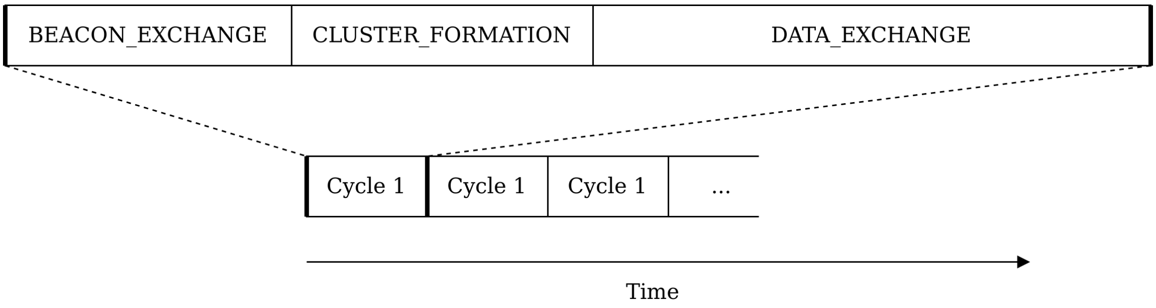

Figure 1: Processes in a Time Cycle

- BEACON_EXCHANGE: The vehicles stay in the STANDALONE state and periodically broadcast their beacons.
- CLUSTER_FORMATION: The vehicles carry out the election and cluster formation procedure. When this process ends, they become either CH or CM.
- DATA_EXCHANGE: Base on the state of the nodes, nodes with have different schemes form sending and forwarding packets.

The duration for each process should be chosen properly to ensure the operation in following steps. Basically, the ultimate purpose of clustering establishment is to support better data exchange within network, therefore, the DATA_EXCHANGE process should cover most of the time in a cycle. BEACON_EXCHANGE and CLUSTER_FORMATION can be considered as overload of the network since there are only information for topology management are exchanged in that interval. The more time that those processes can reduce, the more time we can spend for DATA_EXCHANGE but have to ensure the quality of the outputs.

The detailed description about each process in a cycle is given in the next parts of this section, but before that to be able to build this algorithm, we have made some assumptions:

- All participant in network (including vehicles and RSUs) are equipped with similar on-board uint and have same radio range.
- Cluster's size is limited in a radius of communication range (R)
- There is a RSU after a certain distance
- RSUs can communicate with vehicles and have the ability to access to the core network

3.2 BEACON_EXCHANGE process

In this process, nodes (both vehicles and RSUs) broadcast their information according to a template including:

- ID: The identification number of node
- s: NodeState or state of node, which can be CH/CM/STANDALONE
- ts: timestamp when the data is recorded
- p: Position vector of node at ts
- v: Velocity vector of node at ts
- CID: ID of the CH of the cluster that node belongs

When node i receives a beacon message from j, it extracts the information and calculate the distance to j after an interval, we call this future time δT. If j still in communication range of i, i will save j into its stable neighbor list. Algorithm 1 gives details about this process.

Algorithm 1: Beacon message processing

```
On receiving beacon message from node $j$
if node $j$ is vehicle then
    if CheckOutOfCommunicationRange($j$) == false then
        SaveToNeighborList(B)
    end
else
    if node $j$ is RSU then
        if DistanceTo($j$) < DistanceTo(current closest RSU) then
            SetClosestRSU($j$)
        end
    end
end
```

3.3 CLUSTER_FORMATION process

When this process start, nodes calculate its T_{wait} and wait to announce a Cluster Formation message. If it received a Cluster Formation message from a neighboring node, it stop waiting and became a CM of the sender's cluster and broadcast a new message to inform its neighbors about its new state. If it does not receive any Form Cluster message, when the waiting time is up, it changes its state to CH and broadcast a Cluster Formation message.

Nodes can become a CM if the CH of the cluster it joins is in its neighbor list. During the process, if a node receive a message from its neighborhood which is not the Cluster Formation message (we call this is UpdateStateMessage), it has to update the state of that node in the neighbor list.

Algorithm 2: Cluster Head Election and Cluster Formation

```
When CLUSTER_FORMATION starts
CalculateTwait()
while $T_{wait}$ > 0 do
    if receive a Form Cluster message from node $j$ then
        if neighbor list has node $j$ then
            QuitElection()
            $s_i \leftarrow CM$
            $CID_i \leftarrow ID_j$
            SendUpdateStateMessage()
        end
    else
        Decrement $T_{wait}$
    end
end
$s_i \leftarrow CH$
$CID_i \leftarrow ID_i$
SendClusterFormationMessage()
```

3.4 DATA_EXCHANGE process

After constructing VANETs into clusters, nodes start sending safety and non-safety messages. The communication within a cluster-based network can be divided into two types: intra-cluster and inter-cluster.

In intra-cluster communication, nodes that belong to the same cluster exchange data for each other. If the source node i has a neighboring relationship with the destination node j (j is in neighbor list of i), the packet is directly sent to j. Unless i cannot find j from its neighbor list, i will sent packet to CH, the CH then finds the address of destination node in its neighbor list and forward the packet to the destination.

From our algorithm for CH election, we can see that when node i broadcasts the Form Cluster message, there are two cases which make the rest STANDALONE nodes will not become the CM of i. First case, node receives the Form Cluster (which means it is inside the communication range) but is not a neighbor of i. Second case, nodes which is too far from i so that it cannot receive the Form Cluster message. One of those nodes, will have chances to become the next CH.

However whatever node becomes CH, the connections between two CHs of adjacent clusters are not stable or even do not exist during DATA_EXCHANGE process. Hence, it is hard to deliver a packet across clusters. To overcome this problem, we need the help of RSUs.

When CH realizes that the destination of the packet is not one of its members, it forwards the packet to nearest RSU. As mentioned in 3.1, RSUs are assumed to have access to core network, where all the information about topology of the VANETs is stored. However, the communication environment in the core network is out of this thesis's scope, we made an assumption that after RSU system received the packet, the packet will be delivered to the CH of the cluster containing the destination node by the nearby RSUs. The CH then will forward the packet to the destination.

The data exchange algorithms for CHs are generalized as pseudo-code in Algorithm 3.

Algorithm 3: DATA_EXCHANGE for CH

```
On receiving a Data packet from node $j$ ($p_j$)
if not the packet's destination then
    if neighbor list has the packet's destination node then
        ForwardPacketTo(destination node)
    else
        ForwardPacketTo(current closest RSU)
    end
end
On receiving a Data packet from RSU
if $ID_i$ == destination cluster of the packet then
    if neighbor list has destination node then
        ForwardPacketTo(destination node)
    end
end
```

IV. Tools and Implementations

4.1 NS-3 network simulator

We use NS-3 simulation tools in order to implement the proposed protocol and run the simulation for our scenarios. The NS-3 simulator is a discrete-event network simulator targeted primarily for research and educational use[6]. It provides implementation code execution environment which allows users to simulate the real world network on computer by writing C++ script or Python. NS-3 helps to create virtual nodes (node is a basic computing device abstraction used by NS-3), with the help of various Helper classes, it allows us to install devices, protocol stacks, applications, etc to our nodes.

NS-3 also supports many type of communication such as: PointToPoint, Wireless, Carrier-Send Medium Access (CSMA). Latest NS-3 release maintains an implementation for WAVE (Wireless Access in Vehicular Network) - a system architecture for wireless-based vehicular communications specified by IEEE. We work on version 3.31 of NS-3 simulator, provide a new application model that implements our clustering protocol based on built-in functions from WAVE module.

4.2 Implementations

4.2.1 Designs of classes

In NS-3 the basic abstraction for a user program that generates some activity to be simulated is the application[6]. This abstraction is represented in C++ by the class Application. Here, we extends the built-in Application class into VApplication and RsuApplication, which will manage the behavior of vehicles and Rsu during the cluster cycle time. Figure 2 shows the dependency graph of our applications including some basic functions.

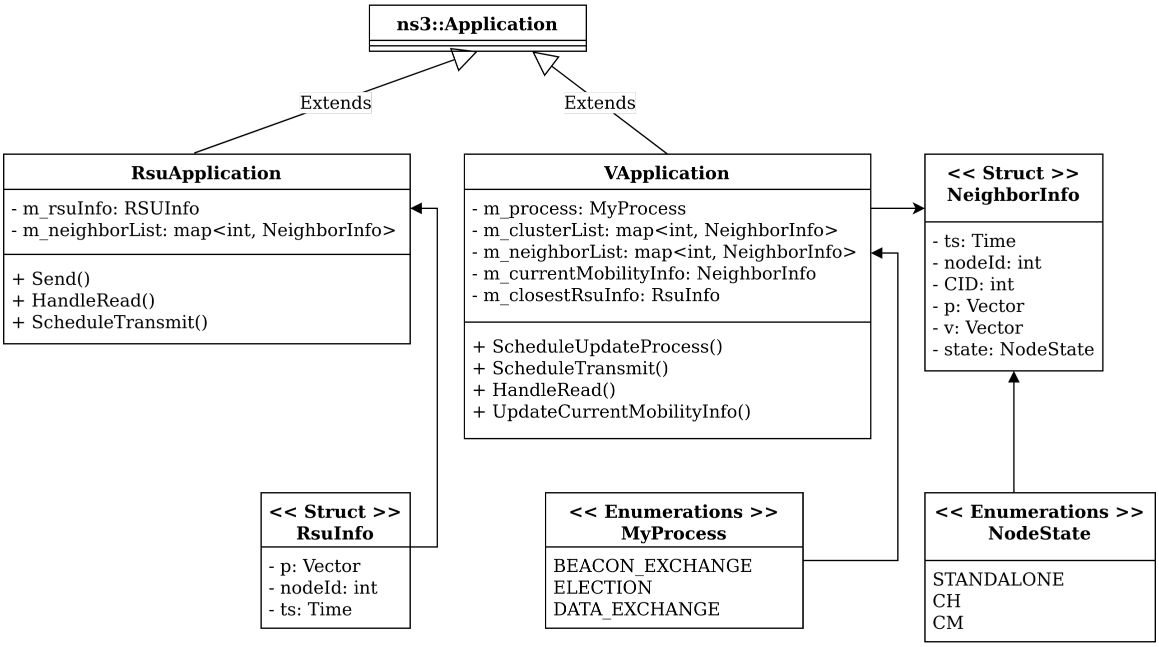

Figure 2: Class designs for VApplication and RsuApplication

Some core functions of VApplication should be mentioned are:

- ScheduleUpdateProcess() which is responsible for switching to new process when the execution time of current process is up and reset the data of the application when it starts a new cycle time.
- ScheduleTransmit() will base on the current process of the time cycle to determine when need to send a packet
- Send() function depends on current state of the node and process to determine which type of packet should be sent. It can call ScheduleTransmit() function after successful sending a packet if the current process require sending information periodically like in BEACON_EXCHANGE.
- HandleRead() function which is called whenever a node receives a packet. It go through the metadata of the packet to get the header type so that it can extract the data embedded in header. Depends on the header and the current state of node, this function will either forward packet, schedule a new send event or changes its state and update its metadata.

There are some specific functions which are not showed in Figure 2. but support for our algorithm.

- CheckOutOfCommunicationRange(): this function will call whenever a node receives a beacons from other nodes to check the ability of a node to become a stable neighbor (added into neighbor list).
- CalculateTwait(): this function is called right at the beginning of CLUSTER_FORMATION process, so that the ScheduleTransmit() function knows when node will send the Cluster Formation packet.

On the other hand, RsuApplication does not directly join the cluster establishment procedure but still keeps a role in data exchange and CH election process. Its operation scheme will not follow the process schedule for a time cycle but still have to manage read and send packet within network.

In addition, we present the implementation of several types of headers via extending the built-in class Header. Packets exchanged in different process and for different purposed will have different header embedded. Those types including:

- FormClusterHeader: CH uses to announce it as a CH and call for neighbor nodes to join its cluster
- VBeaconHeader: Vehicles use to send beacon message in BEACON_EXCHANGE process and in CH election to inform theirs new states
- RsuBeaconHeader: RSUs use to broadcast their information in network
- DataHeader: used for data packet exchanged in DATA_EXCHANGE process Figure 3 presents the designs of our headers, each will override the Serialize() and Deserialize() method as well as provide necessary getter and setter for its data.

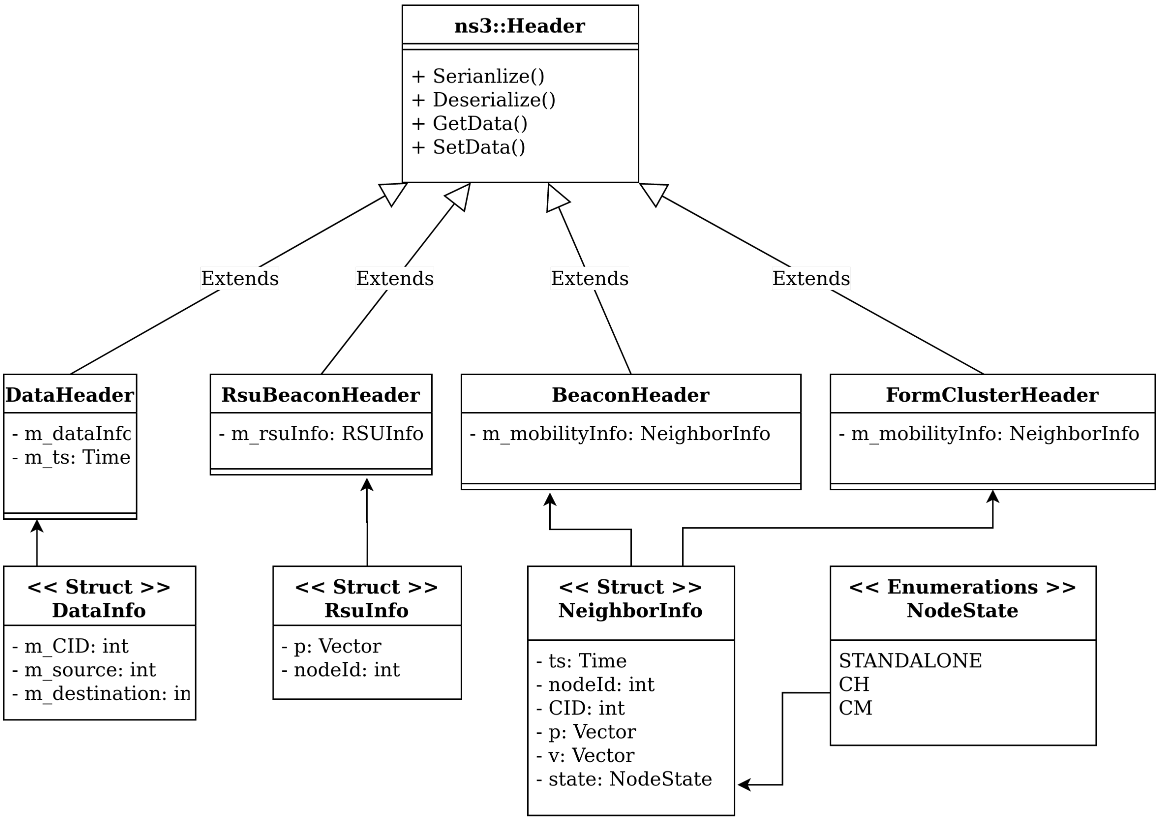

Figure 3: Designs for headers

V. Evaluations and Discussions

In this section, we provide evaluation of our clustering protocol based on how much it can improve the connectivity of the vehicular network. As mentioned before, the connectivity is defined by how many vehicles within a specified area that a vehicle has connection with. However, since we are working with a network with high mobility like VANET, it is necessary to consider the stability of each connection during the whole data exchange duration also.

In order to evaluate the performance of our algorithm, we have studied how the Packet Delivery Ratio (PDR - the ratio of packets successfully received to the total sent) varies when increasing the initials node density in two scenarios: stable and non-stable scenarios. For each scenario, we schedule several experiments with different initial positions or velocity of nodes, measure the PDR and compare to the PDR when no protocol applied.

In our experiment, when the DATA_EXCHANGE process starts, we schedule a node to continuously send packets to each other node in the network until the end of DATA_EXCHANGE process and collect the number of packets received at each destination node. Then we can calculate the PDR for each V2V communication in the network. For the PDR of the whole network, we take the average of all V2V communications.

For both scenarios, we setup simulation with nodes are initially distributed on a three-lane straight road 100m long at maximum and the vertical distance between two consecutive lane is 4m. The number of nodes increases from 5 to 45. The original positions and velocities of nodes are generated using an uniform random variable stream. Velocities of nodes are unchanged during the simulation time. Table 1 illustrates the common configuration of the basic parameters of the scenarios. Since nodes are moving, in order to stay up to date with the closest RSU, several RSUs are sett up every 100m.

Table 1: Common configuration parameters of the clustering scenarios

Parameter	Value
Position range	[0m, 100m]
Mobility Model	Constant Velocity

Parameter	Value
Simulation Time	33s
Number of Vehicles	5/10/15/20/25/30/35/40/45
Distance between two adjacent RSUs	100m

Stable Scenario

Stable does not mean nodes do not move in this scenario but the relative velocity between nodes is so small that it does not affect the topology of the network in long term. Figure 3 shows the results of simulation for scenario 1. Since the position range of nodes in our scenario is limited to 2 communication ranges, the result of clustering protocol gives us 1 cluster no matter how many nodes there are.

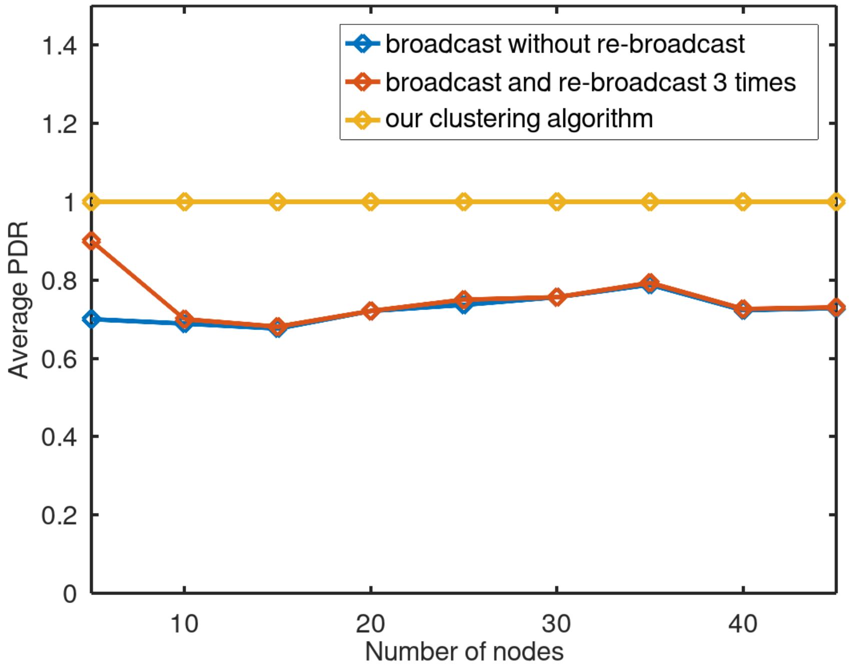

Figure 4: Stable Scenario

In this situation, every node can talk to each other and the PDR keeps unchanged when we increase the number of nodes in our pre-definced area. The PDR measured from blind-broadcast case, in contrast, varies with the changed of node density. The re-broadcast case only helps improve the PDR when there are few nodes in the network. With higher number of vehicles, re-broadcasting shows no effect on the PDR performance. This can be understandable because the more nodes, the more re-broadcasting in the network, hence, resulting in lack of medium resources and more medium access collisions.

Non-stable Scenario

In non-stable scenario, we consider the mobility of the nodes in simulation. Each node has its velocity generated from an uniform random stream in range [10.3535, 11.8686] m/s. However, nodes still move in the same directions. Figure 5 shows the results of simulation for scenario 2. Compare to the first scenario, the PDR of the whole network decrease, how ever the clustering protocol still has better performance over the broadcast and re-broadcast case.

In this scenario, the PDR decreases when we add more nodes to the network. The loss in PDR is the result of disconnection between two nodes that are not within a communication range and not in the same cluster. Nodes that are in the same cluster still can talk to each other, however the inter-cluster communication is carried out only between two neighboring nodes by direct communication.

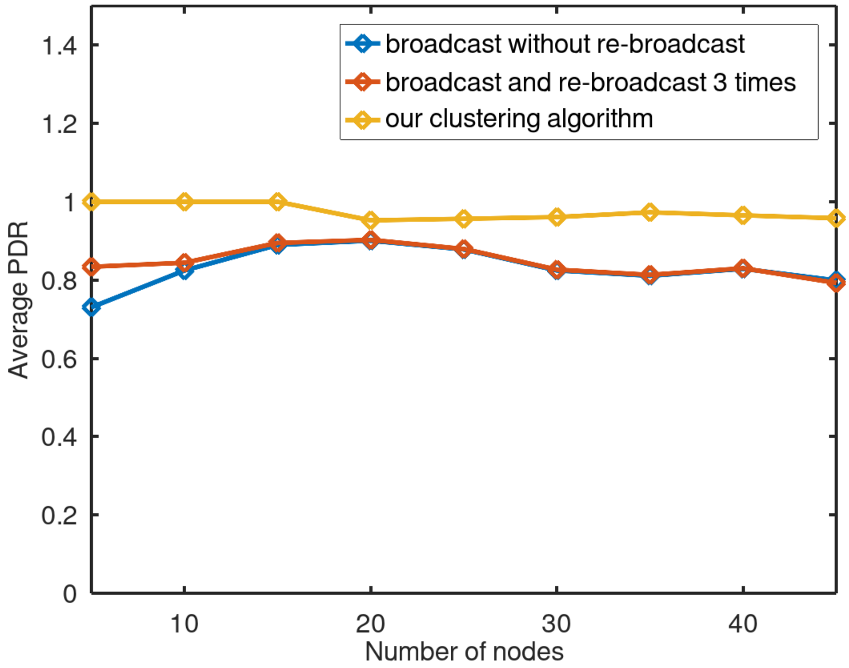

Figure 5: Non-stable Scenario

Discussions

So far, from the two scenarios, we have studied the relationship between the performance of our clustering protocol and the node density and the mobility of the network. According to scenario 1, the node density of a cluster does not affect the inner communications of that cluster. At any time, there is a way to establish a connection between any two nodes in the same cluster.

However, the scenario is limited at point-to-point communication. In a more practical scenario, when we take into account the packet transmission to multiple destinations and the increment in demand of data exchange among CMs, there is a threshold when the CH cannot handle indirect communications or there is no more medium resource.

Moreover, the communication range of nodes is an estimated value. In practice, this value is not fixed by the influence of the terrain as well as the electromagnetic effect. Hence, in a critical case, there is a chance of packet loss in the intra-cluster communication. Our scenarios, by far, haven't detected any packet loss due to this.

Scenario 2, on the other hand, provides the relationship between connectivity and mobility. When the network is more dynamic, clustering protocol establishes two or more clusters. When we add more nodes to the simulation, the algorithm distributes them to different clusters, whether this makes the PDR rises or falls depends on the mobility of the new nodes in network but in general, we can see an inverse trend between PDR and the number of clusters in the network.

Table 2 describes the output of clustering protocol in one experiment when we increase the number of vehicle in the network. For example, when there are 20 vehicles on the road, two cluster are established with node 6 and 2 are CHs and the size of those cluster are 19 and 1 correspondingly.

Table 2: Clusters formed by clustering protocol in scenario 2

Number of Vehicles	Number of Clusters	Cluster List	Cluster distributions
5	1	[3]	[5]
10	1	[3]	[10]
15	1	[3]	[15]
20	2	[6, 2]	[19, 1]
25	2	[3, 17]	[24, 1]
30	2	[2, 6]	[1, 29]
35	3	[2, 6, 30]	[1, 33, 1]
40	3	[2, 6, 30]	[1, 38, 1]
45	3	[2, 6, 30]	[1, 43, 1]

Theoretically, the wider the velocity range and initial position range, the more the number of clusters formed after each cycle. All the above result, are measured in the first cycle time of simulation. The changes in number of clusters is clearly shown if we run the simulation for several cycles and higher velocity range. Figure 6 shows how the number of clusters increases after 3 first cycle time with each cycle lasts 33 seconds.

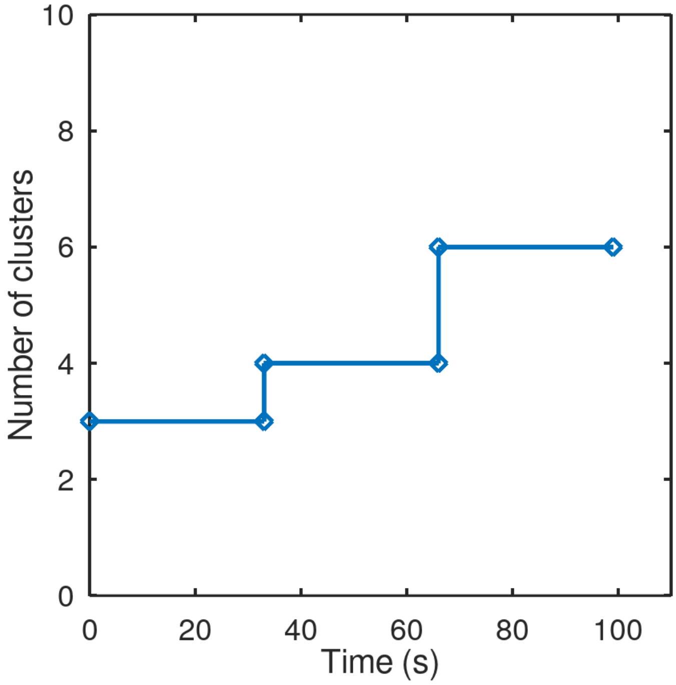

Figure 6: Number of clusters created in the first 100 seconds with cycle time equals 33 seconds

With the purpose of keeping a high connectivity within such a dynamic network like VANET, our protocol should be able to keep up with the rapid change in topology. In order to do that, some specific parameters and formula in our protocol (duration for each process, criteria for adding node to neighbor list, $T_{wait} formula, …) have to be selected properly.

Here we examined the influence of a cycle time by running the experiment again with cycle times are 13 seconds and 23 seconds respectively. We kept the interval of BEACON_EXCHANGE and CLUSTER_FORMATION unchanged and only modified the operation time in DATA_EXCHANGE. The velocity range keeps the same for Figure 6 and 7 but wider than the previous scenarios in order to indicate the change in network topology more clearly.

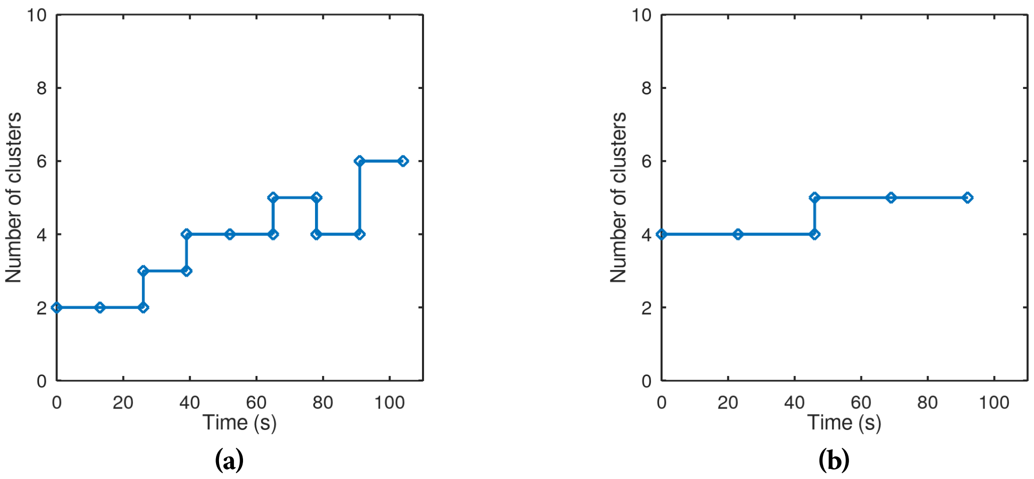

Figure 7: Number of clusters created in the first 100 seconds of simulation with cycle time is (a) 13 seconds and (b) 23 seconds

By setting cycle time to 23 seconds, we can reduce the time that network has to operate at 6 clusters from 33 seconds to 8s. Choosing the 13-second cycle have longer duration with 6 clusters but it results in the first 2 cycles (26 seconds) operating at only 2 clusters, which may give a higher connectivity.

To conclude, choosing higher cycle time (which means more focusing on exchanging data than updating the network topology) will make the protocol to return higher number of clusters because there is fewer nodes satisfying the criteria to be added into neighbor list. This make our network scattered and more depends on inter-cluster communication, hence, overloading is unavoidable.

Lower value of cycle time can reduce the chance of high cluster number situation but the efficiency of data transmission within VANET will be reduced if the BEACON_EXCHANGE and CLUSTER_FORMATION keep repeating unnecessarily every short interval. However, to know how actually this idea can improve the connectivity of the network, further evaluation should be conducted with carefully consider to the trade offs.

VI. Conclusion and Future work

To conclude, we proposed a protocol for clustering establishment in vehicular network with a focus on increasing connectivity in order to support safety application. We implemented the protocol in ns3 simulation system and run the simulation for 2 scenarios and evaluate its performance. Our scenarios are simple, in order to evaluate the new protocol more comprehensively, simulations should be carried out with more practical scenarios, concerning environment factors, noises, power transmit, multichannel communication, medium access, etc…

From the analysed result, there are ways to improve the protocol by modifying some parameters like cycle time, duration of each process, T_{wait} formula, or the criteria to become neighboring nodes.

So far, the most challenge is neither the election algorithm nor beacon exchange part, but in processing the data exchange. As mentioned above, our implementation is lack of inter-cluster communication. Nodes can only communicate with others in adjacent clusters if they are near the common borders. The performance of this protocol would be highly improved if a full data exchange algorithm (when RSUs play a critical role in delivering packets across clusters) is developed.

Appendix

A. Waiting time calculation in CLUSTER_FORMATION process

Given a vehicle node i which has neighbor list N_i, distance to closest RSU d_RSU and communication range R, the waiting time of node in election process is calculated as following:

$$T_{wait} = (1 - connectivity\_index)T_{wait\_max}$$

where:

- $connectivity\_index = \frac{distanceindex + centralityindex}{2}$
- $distance\_index = \frac{R - d_{RSU}}{R}$
- $centrality\_index = 1 - \frac{1}{sizeof(N_i)}$

References

[1] Shrestha, Rakesh et al. “Challenges of Future VANET and Cloud-Based Approaches.” *Wireless Communications and Mobile Computing* 2018 (2018): 5603518:1-5603518:15.

[2] M. Gerla and J. T. Tsai, “Multiuser, mobile, multimedia radio network.” *Wireless Networks*, vol. 1, pp. 255-265, October 1995.

[3] RA Santos, RM Edwards, NL Seed. “Inter vehicular data exchange between fast moving road traffic using ad-hoc cluster based location algorithm and 802.11b direct sequence spread spectrum radio.” *Post-Graduate Networking Conference* 2003.

[4] Z. Y. Rawashdeh. “A novel algorithm to form stable clusters in vehicular ad hoc networks on highways.” *EURASIP Journal on Wireless Communications and Networking* January 2012.

[5] Arkian, Hamidreza & Ebrahimi Atani, Reza & Pourkhalili, Atefe & Kamali, Saman. “Cluster-based traffic information generalization in Vehicular Ad-hoc Networks.” *Vehicular Communications* vol. 1 October 2014.

[6] ns-3 project, “NS3 Tutorial Release ns-3-dev”. Jun 2020